

LAW OFFICES  
**McGuireWoods LLP**  
1750 TYSONS BOULEVARD, SUITE 1800  
MCLEAN, VIRGINIA 22102

APPLICATION  
FOR  
UNITED STATES  
LETTERS PATENT

Applicants: Yiming Ye, Stephen J. Boies, Paul Y.  
Huang and John K. Tsotsos  
For: AUTOMATICALLY DETERMINING THE  
AWARENESS SETTINGS AMONG PEOPLE  
IN DISTRIBUTED WORKING ENVIRONMENT  
Docket No.: YOR920010322US1

# AUTOMATICALLY DETERMINING THE AWARENESS SETTINGS AMONG PEOPLE IN DISTRIBUTED WORKING ENVIRONMENT

## DESCRIPTION

5

### BACKGROUND OF THE INVENTION

#### *Field of the Invention*

The present invention generally relates to collaborative computing and, more particularly, to a way to automatically determine the awareness settings among people in a distributed working environment.

10

#### *Background Description*

Collaborative computing is a shared computing environment or application that facilitates communication and teamwork among groups of people.

15      Displaying common data on multiple computers is one type of collaborative work support system. U.S. Patent No. 5,996,002 to Katsurabayashi et al. for “Collaborative Work Support System and Method to Facilitate the Process of Discussion in a Meeting Using a Shared Window” discloses a system that includes shared data that is displayed on each computer, individual data that is individually displayed on any of the 20 computer selected by the data owner, and a manager for managing data storage. A similar system that displays shared data on multiple computers is disclosed in U.S. Patent No. 5,872,924 to Nakayama et al. for “Collaborative

Work Support System" in which a limited amount of data is displayed according to multiple attributes setting in the shared windows.

A detail log is a simple way to provide user activity information. U.S. Patent No. 5,008,853 to Bly et al. for "Representation of Collaborative Multi-user Activities Relative to Shared Structured Data Objects in a Networked Workstation Environment" proposed a multi-user collaborative system in which the contents as well as the current status of other user activity can be concurrently accessed by different users. The WYSIWIS (What You See Is What I See) user interface representation includes an ordered listing of such entries that are maintained by the structured data object and various attributes of each listed entry; inter alia, the type and class of entry, the revision number of the shared structured data object, the number of pages and revision number of each structured data object entry, the date of creation and last revision of each such entry, whether an entry can be accessed by a user and, if not, who has prevented such access to prevent concurrent editing, whether a local instance of an entry is present on a user's system, and a provision for miscellaneous notes or comments relative to each entry for view by other users.

Visual representation of other users is a standard way of monitoring other's activity. U.S. Patent No. 5,793,365 to Tang et al. for "System and Method Providing a Computer User Interface Enabling Access to Distributed Workgroup Members" discloses a system that uses a user interface to display visual representations of selected other users in the workgroup. The visual representations are frequently updated to indicate the activity level of these users. An encounter awareness system detects the presence of other users who are doing similar tasks. U.S. Patent No. 4,974,173 to Stefik et al. for "Small-scale Workspace Representations Indicating Activities by Other Users" proposes a computer system and method that provide networked computer users with information about which other users are task proximate to the user,

thereby facilitating spontaneous communications regarding task-related, or other, issues. Task proximity to other users may change as the user context switches between applications, and the user interface window is updated accordingly. Task proximity is determined individually by different  
5 applications.

Discourse manager has been used to promote effective collaboration between a user and a collaborative computer agent. U.S. Patent No. 5,819,243 to Rich et al. for "System with Collaborative Interface Agent" suggested a system which operates according to a theory of collaborative discourse  
10 between humans, with the computer agent playing the same role as a human collaborator. The discourse manager includes a memory in which application-specific recipes are stored and a memory in which the discourse state is stored. Each recipe specifies a set of actions or sub-tasks, which are performed to achieve an objective. The discourse state includes structures to track the  
15 agent's and user's current objectives, a selected recipe for each objective, and completed steps in each recipe. During operation of the discourse manager, user actions and communications are interpreted according to how they relate to the current discourse state. The manager also generates an agenda of expected communications, which is presented to the user as a menu, obviating the need for the natural language understanding by the agent.  
20

Prior work in collaborative computing can provide awareness among group members through various communication channels, such as video, audio, graphical user interface (GUI), etc. However, the prior work requires users to adjust the communication channel and the degree of communications.  
25 For example, when a person leaves his or her office, he or she has to manually turn the audio and video off if it is not needed. This is not convenient.

## SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a way to automatically adjust communication channels among users based on users' current states detected by various sensing devices.

According to the invention, there is provided a collaboration system that includes an awareness system for evaluating, monitoring, and controlling, in real-time, the collaboration environment by having events and occurrences with properties. The awareness monitoring system includes (1) input sensors for receiving real-time data produced by the event and (2) an elastic spring energy model for automatically adjusting a distance that is according to a level of privacy desired by individual users and a need of the collaborative project to have some shared information about individual user activities.

## BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages will be better understood from the following detailed description of a preferred embodiment of the invention with reference to the drawings, in which:

Figure 1 is a block diagram of one preferred embodiment of the system;

Figure 2 is a block diagram showing a user environment;

Figure 3 is a diagram showing different channels for communication between agents;

Figure 4 is a flow diagram showing the process of the system setup procedure;

Figure 5 is a flow diagram showing the process of the awareness network build up process;

Figure 6 is a network diagram graphically illustrating the awareness

network;

Figure 7 is a flow diagram showing the process of obtaining the ideal distance for preparing on-line activities;

5       Figure 8 is a flow diagram showing the process of obtaining ideal distance with respect to the agent at hand;

Figure 9 is a flow diagram showing the process of obtaining ideal distance with respect to the organization;

Figure 10 is a flow diagram showing the process of obtaining ideal distance with respect to the other agents;

10      Figure 11 is a flow diagram showing the process of obtaining ideal distance with respect to the current task;

Figure 12 is a flow diagram showing the process of obtaining spring constants for the system;

15      Figure 13 is a flow diagram showing the process of obtaining spring constants for the agent's own user;

Figure 14 is a flow diagram showing the process of obtaining spring constants for the organization;

Figure 15 is a flow diagram showing the process of obtaining spring constants for other agents;

20      Figure 16 is a flow diagram showing the process of obtaining spring constants for the current task;

Figure 17 is a flow diagram showing the process of generating the Event-Weight table;

25      Figure 18 is a flow diagram showing the process of obtaining values from the Event-Weight table;

Figure 19 is a flow diagram showing the working process of the system;

Figure 20 is a flow diagram showing the process of online distance selection for a given event;

Figure 21 is a flow diagram showing the process of best distance calculation;

Figure 22 is a flow diagram showing the process of obtaining the values of the back-to-ideal potential energy matrix;

5       Figure 23 is a flow diagram showing the process of obtaining the values of the elements of the back-to-ideal potential energy matrix;

Figure 24 is a flow diagram showing the process of obtaining the back-to-ideal potential energy vector;

10      Figure 25 is a flow diagram showing the process of obtaining the back-to-ideal potential energy vector from  $i$  to  $j$  with respect to agent  $j$ ; and

Figure 26 is a flow diagram showing the process of obtaining the back-to-ideal potential energy vector from  $i$  to  $j$  with respect to a given task  $t_q$ .

#### **DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT OF THE INVENTION**

15      Referring now to the drawings, and more particularly to Figure 1, there is shown one preferred embodiment of the system. Blocks 101<sub>1</sub> to 101<sub>5</sub> represent the environments of single users. Blocks 105<sub>1</sub> to 105<sub>5</sub> represent agents for respective users. Lines 110<sub>1</sub> to 110<sub>10</sub> represent the communication channels between two agents.

20      Figure 2 shows in more detail a user environment. 201 refers to those devices surrounding a user. These might include, for example, a personal computer 202, a printer 203, and a cell phone 204, among other devices. At the center of the diagram, 205 refers to the user. Through communications with agents of other users, information pertaining to a current user can be delivered to other users, and vice versa.

25      Figure 3 is a diagram showing different channels for communication between agents. 301 represents a video channel that can transmit video signals

from/to users. 305 is the audio channel. 315 is the channel for orders. 315 is the channel that transmit the typing activities of users. 355 represents the channel for users' positions. These positions can be detected by many devices, such as a Global Position System, etc.

Figure 4 is a flow diagram showing the system setup procedure. In function block 401, the input device is determined. Input devices are those devices that can sense the user. For example, a keyboard is an input device, a camera is an input device, a seismometer is an input device, a microphone is an input device. There are many examples of input devices. Function block 405 determines the output devices. For example, a screen is an output device, a speaker is an output device, etc. Some devices are both input and output devices, for example, a touch screen is both an input and output device. In function block 415, all the possible channels for each user are determined. Suppose that there are a total of  $m$  different channels. A channel refers to an independent way of representing an user's activities to the other users. For example, a video channel is a channel for presenting an user's activities. An input device such as a camera can grab the images and an output device such as a computer screen can output the activities. The video channel can be one channel for the system. Another channel example is the audio channel. A microphone can grab voice signals and a speaker can send the signal to relevant user. Function block 455 determines the distances for each channel. The distance refers to how clear the receiver can receive the corresponding signal of the other party. For example, in a video channel, the receiver may get a very clear video image or it may get a fuzzy image or it may only get the image after processing or the transmitting rate might be too high or low, etc. The task of this block is to identify the different degrees of clearness of the video image transmitted and then assign to these different degrees a distance number. The higher the distance number, the lower the quality of the signals transmitted through this channel. In a practical system, we can simply assign

all the possible different choices to a number between 1 and 100. Where when the distance equals 1, the video signal will be the best one that can be possibly transmitted. When the distance equals 100, the quality of the signal will be the worst one that can be transmitted.

5           Figure 5 is a flow diagram showing the awareness network build up process. In function block 501, all the peers within the group are identified. This will define the range of the system. For purposes of illustration, assume  $n$  is the total number of peers in the system. Function block 505 represents each peer by a node. Thus, we have a total of  $n$  nodes available in the awareness network. In function block 515, the communication channels among the peers is specified. For each pair of peers  $i$  and  $j$ , we have  $m$  direct links from node  $i$  to node  $j$ , and  $m$  direct links from node  $j$  to node  $i$ . Each link represent a communication channel. A link from node  $i$  to node  $j$  refers to the corresponding communication channel that used to deliver the signals for user  $i$  to be presented to user  $j$ . In function block 555, for each link, the total number of choices/distances that can be used is specified. This number comes 10       from function block 455 in Figure 4.

15

20           Figure 6 is a diagram shows a graphical description of the awareness network based on the embodiment of Figure 1. There are five nodes in the awareness network, and there are only one channel for each peer of nodes. The status of a awareness network can be represented by an awareness matrix  $\mathbf{A}$ :

$$\mathbf{A}(k) = \begin{bmatrix} a_{11}(k) & \cdots & a_{1n}(k) \\ \vdots & \ddots & \vdots \\ a_{nl}(k) & \cdots & a_{mn}(k) \end{bmatrix},$$

where  $a_{ij}(k)$  gives the value of the distance from user  $i$  to user  $j$  with respect to channel  $k$  for the awareness network. Here  $k$  can be any value from 1... $m$ .

25           When  $i=j$ , the value of  $a_{ii} = 0$ .

In addition to the matrix representing the current state of the awareness network, we also use another matrix to represent ideal distances with respect to different channels and different parties. The ideal distance that an agent want to provide to other agents is represented by matrix  $\mathbf{S}$ :

5

$$\mathbf{S}(k,e) = \begin{bmatrix} s_{11}(k,e) & \cdots & s_{1n}(k,e) \\ \vdots & \ddots & \vdots \\ s_{n1}(k,e) & \cdots & s_{nn}(k,e) \end{bmatrix},$$

where  $s_{ij}(k,e)$  gives the value of the ideal distance that user  $i$  wants to provide to user  $j$  with respect to channel  $k$  for a given event  $e$ . Here  $k$  can be any value from  $1 \dots m$ . When  $i=j$ , the value of  $s_{ii} = 0$ . This is the only place that events  $e$  matters.

10

The ideal distance that the organization want is represented by matrix  $\mathbf{G}$ :

$$\mathbf{G}(k) = \begin{bmatrix} g_{11}(k) & \cdots & g_{1n}(k) \\ \vdots & \ddots & \vdots \\ g_{n1}(k) & \cdots & g_{nn}(k) \end{bmatrix},$$

15

where  $g_{ij}(k)$  gives the value of the ideal distance that the organization wants user  $i$  to provide to user  $j$  with respect to channel  $k$ . Here  $k$  can be any value from  $1 \dots m$ . When  $i=j$ , the value of  $g_{ii} = 0$ . Basically, the organization of the team may have some special requirement, these requirements will be one of the factors that might influence the final decisions of an agent.

The ideal distance that the other agents want is represented by matrix  $\mathbf{O}$ :

$$\mathbf{O}(k) = \begin{bmatrix} o_{11}(k) & \cdots & o_{1n}(k) \\ \vdots & \ddots & \vdots \\ o_{n1}(k) & \cdots & o_{nn}(k) \end{bmatrix},$$

where  $o_{ij}(k)$  gives the value of the ideal distance that the user  $j$  wants user  $i$  to provide to him with respect to channel  $k$ . Here  $k$  can be any value from  $1 \dots m$ . When  $i=j$ , the value of  $o_{ii}=0$ . Basically, this is the other agent's requirement.

5 For a given agent, it must consider all the other agents' requirements.

The ideal distance required by the given task is represented by matrix

**T:**

$$\mathbf{T}(k) = \begin{bmatrix} t_{11}(k) & \cdots & t_{1n}(k) \\ \vdots & \ddots & \vdots \\ t_{n1}(k) & \cdots & t_{nn}(k) \end{bmatrix},$$

10 where  $t_{ij}(k)$  gives the value of the ideal distance that the current task wants user  $i$  to provide to user  $j$  with respect to channel  $k$  for a given event  $e$ . Here  $k$  can be any value from  $1 \dots m$ . When  $i=j$ , the value of  $t_{ii}=0$ . Basically, different tasks the team is involved in will have different requirements. For example, if the team is having a brain storming session, then the requirement for video is very high, while if all the members of the team is just working on their own  
15 stuff, then a video connection may not even be needed. The requirement of the task is independent of the events happening on each user's site.

Figure 7 is a flow diagram showing the process of obtaining the ideal distance for preparing on-line activities. In function block 701, the values for matrix **S** are obtained. Function block 705 obtains the values for matrix **G**.  
20 Function block 715 obtains the values for matrix **O**. Function block 755 obtains the values for matrix **T**.

Figure 8 is a flow diagram showing the process of obtaining an ideal distance with respect to the agent at hand. It actually fills one row of the matrix  $\mathbf{S}$ . Function block 801 selects the first channel to set the ideal distance. Function block 805 chooses the first agent other than the current agent in question. A processing loop is then entered at function block 815 where the ideal distance for the corresponding channel is set. It gives the value of  $S_{a,i}(k,e)$ , where  $a$  is the agent for the current user in question. In function block 855, the next agent to consider is chosen. A determination is made in decision block 865 to determine if all the agents are considered. If so, the process goes to the next channel in function block 875; otherwise, the process loops back to function block 815 to choose the next agent. A determination is made in decision block 895 as to whether all the channels are considered and, if not, this process is repeated for all the possible events.

Figure 9 is a flow diagram showing the process of obtaining an ideal distance with respect to the organization. In function block 901, the first channel is chosen. The process enters a processing loop at function block 905 where the initial agents are set. A nested processing loop is entered at function block 915 where the values of  $g_{i,j}(k)$  are obtained. Function block 955 moves to the next distance start agent. Decision block 965 checks whether all the start agents are covered. If not, the process loops back to function block 915 to cover the next distance agent; otherwise, the process goes to function block 971 where the start agent is re-set to 1. Function block 975 moves to the next distance ending agent. Decision block 977 checks whether all the distance ending agents are covered. If not, the process loops back to function block 915 to cover the new distance start agent and the new distance ending agent; otherwise, the process goes to function block 979 to check the next channel. Decision block 995 checks whether all the channels are covered. If not, the process loops back to function block 905 to cover all the distance start agents and distance ending agents; otherwise, the process is finished.

Figure 10 is a flow diagram showing the process of obtaining an ideal distance with respect to the other agents. In function block 1001, the initial channel is set, and in function block 1005 the initial agent is set. A processing loop is entered in function block 1007 where the value of  $o_{a_i}(k)$  is set.

5 Function block 1015 chooses the next agent. A determination is made in decision block 1017 as to whether all agents are covered. If so, in function block 1019, the next channel is chosen; otherwise, the process loops back to function block 1007. Then, in decision block 1055, a determination is made as to whether all channels are covered. If not, the process loops back to function block 1007; otherwise, the process is terminated.

10

Figure 11 is a flow diagram showing the process of obtaining an ideal distance with respect to the current task. In function block 1101, the first channel is chosen. Function block 1105 sets the initial agents. Block 1115 sets the value of  $t_{i_j}(k)$ . Function block 1117 moves to the next distance start agent. Decision block 1151 checks whether all the start agents are covered. If not, the process loops back to function block 1115 to cover the next distance ending agent; otherwise, the process goes to function block 1155 where the start agent is re-set to 1. Function block 1157 moves to the next distance ending agent. Decision block 1175 checks whether all the distance ending agents are covered. If not, the process loops back to function block 1115 to cover the new distance start agent and the new distance ending agent.

15

20

To proceed with other drawing figures, we need to specify several other matrices. Since we model the system as a spring system, we need to specify the constants for calculation. These constants will be used in calculating the best distances.

25

The spring constant that an agent want to provide to other agents is represented by matrix  $\mathbf{K}_S$ :

$$\mathbf{K\_S}(k) = \begin{bmatrix} k\_s_{11}(k) & \dots & k\_s_{1n}(k) \\ \vdots & \ddots & \vdots \\ k\_s_{n1}(k) & \dots & k\_s_{nn}(k) \end{bmatrix},$$

where  $k\_s_{ij}(k)$  gives the user  $i$ 's spring constant for channel  $k$  from  $i$  to  $j$ . Here  $k$  can be any value from  $1\dots m$ . When  $i=j$ , the value of  $k\_s_{ii} = 0$ .

The spring constants that the organization want is represented by  
5 matrix  $\mathbf{K\_G}$ :

$$\mathbf{K\_G}(k) = \begin{bmatrix} k\_g_{11}(k) & \dots & k\_g_{1n}(k) \\ \vdots & \ddots & \vdots \\ k\_g_{n1}(k) & \dots & k\_g_{nn}(k) \end{bmatrix},$$

where  $k\_g_{ij}(k)$  gives the value of the constant for channel  $k$  from  $i$  to  $j$  with  
respect to the organization. Here  $k$  can be any value from  $1\dots m$ . When  $i=j$ , the  
value of  $k\_g_{ii} = 0$ . For a given agent  $i$ , it can only access the  $i$ -th row of the  
10 matrix. Basically, the organization of the team may have some special  
requirement, these requirements will be one of the factors that might influence  
the final decisions of an agent.

The spring constants that the other agents want is represented by  
matrix  $\mathbf{K\_O}$ :

$$15 \quad \mathbf{K\_O}(k) = \begin{bmatrix} k\_o_{11}(k) & \dots & k\_o_{1n}(k) \\ \vdots & \ddots & \vdots \\ k\_o_{n1}(k) & \dots & k\_o_{nn}(k) \end{bmatrix},$$

where  $k\_o_{ij}(k)$  gives the value of the spring constant for channel  $k$  with regard  
to the issue that user  $j$  want user  $i$  to provide to him. Here  $k$  can be any value

from 1... $m$ . When  $i=j$ , the value of  $k\_o_{ji} = 0$ . Agent  $i$  can only access the  $i$ -th row. Basically, this is the other agent's requirement. For a given agent, it must consider all the other agents' requirements.

5 The spring constant with respect to the given task is represented by matrix  $\mathbf{K\_T}$ :

$$\mathbf{K\_T}(k) = \begin{bmatrix} k\_t_{11}(k) & \dots & k\_t_{1n}(k) \\ \vdots & \ddots & \vdots \\ k\_t_{n1}(k) & \dots & k\_t_{nn}(k) \end{bmatrix},$$

where  $k\_t_{ij}(k)$  gives the value of the spring constant for channel  $k$  from user  $i$  to user  $j$ . Here  $k$  can be any value from 1... $m$ . When  $i=j$ , the value of  $k\_g_{ii} = 0$ . Agent  $i$  can only access the  $i$ -th row.

10 Figure 12 is a flow diagram showing the process of obtaining spring constants for the system. In function block 1201, the values of  $\mathbf{K\_S}$  are obtained. In function block 1205, the values of  $\mathbf{K\_G}$  are obtained. In function block 1215, the values of  $\mathbf{K\_O}$  are obtained. Finally, in function block 1255, the values of  $\mathbf{K\_T}$  are obtained.

15 Figure 13 is a flow diagram showing the process of obtaining spring constants for the agent's own user. It actually fills one row of the matrix  $\mathbf{K\_S}$ . In function block 1301, the first channel to set the spring constant is selected. Function block 1305 chooses the first agent other than the current agent in question. A processing loop is entered at function block 1315 where the spring constant for the corresponding channel is set. It gives the value of  $K\_S_a(k,e)$ , where  $a$  is the agent for the current user in question. In function block 1355, the next agent to consider is chosen. A determination is made in decision block 1365 as to whether all the agents are considered and, if so, the process goes to the next channel; otherwise, the process loops back to function block 20 1315 to choose the next agent. Decision block 1395 checks whether all the 25 1315 to choose the next agent. Decision block 1395 checks whether all the

channels are considered.

Figure 14 is a flow diagram showing the process of obtaining spring constants for the organization. In function block 1401, the constant for first channel is chosen. Function block 1405 sets the initial agents. A processing loop is entered at function block 1415 which obtains the values of  $k_{g_{ij}}(k)$ .  
 5 Function block 1451 moves to the next distance start agent. Decision block 1455 checks whether all the start agents are covered. If not, the process loops back to function block 1415 to cover the next distance start agent; otherwise, the process goes to function block 1465 where the start agent is re-set to 1.  
 10 Function block 1471 moves to the next distance ending agent. Decision block 1475 checks whether all the distance ending agents are covered. If not, the process loops back to function block 1415 to cover the new distance start agent and new distance ending agent; otherwise, the process is finished.

Figure 15 is a flow diagram showing the process of obtaining spring constants for other agents. Function block 1501 sets the initial channel, and function block 1505 sets the initial agent. A processing loop is entered at function block 1507 where the value of  $k_{o_{ai}}(k)$  is set. In function block 1515, the next agent is chosen. Decision block 1517 checks whether all agents are covered and, if not, the process loops back to function block 1507. Function block 1519 chooses the next channel.  
 15  
 20

Figure 16 is a flow diagram showing the process of obtaining spring constants for the current task. In function block 1601, the first channel is chosen. Function block 1605 sets the initial agents. A processing loop is entered at function block 1615 where the value of  $k_{t_{ij}}(k)$  is set. Function block 1617 moves to the next distance start agent. Decision block 1651 checks whether all the start agents are covered. If not, the process loops back to function block 1615 to cover the next distance start agent; otherwise, the process goes to function block 1655 where the start agent is re-set to 1.  
 25 Function block 1657 moves to the next distance ending agent. Decision block

1675 checks whether all the distance ending agents are covered. If not, the process loops back to function block 1615 to cover the new distance starting agent and the new distance ending agent; otherwise the process is finished.

To illustrate the emphasis of different factors for different agents, we  
 5 use weight matrices to represent those weights. For different agents, the current agent at hand will have different weights. The weights are channel free, because the channel preference is encoded by spring constants.

The weights that related to agents' own preference is represented by matrix  $\mathbf{W\_S}$ :

$$10 \quad \mathbf{W\_S}(k) = \begin{bmatrix} w\_s_{11}(k) & \dots & w\_s_{1n}(k) \\ \vdots & \ddots & \vdots \\ w\_s_{n1}(k) & \dots & w\_s_{nn}(k) \end{bmatrix},$$

where  $w\_s_{ij}$  gives the value of the weight that agent  $i$  assigned to the distance from  $i$  to  $j$  with considering its user's own need. When  $i=j$ , the value of  $w\_s_{ii} = 0$ . Please note that each row represent the needs of each user. Agent  $i$  can only access the  $i$ -th row.

15 The weights that related to an agent's idea on the importance of the organization is represented by matrix  $\mathbf{W\_G}$ :

$$\mathbf{W\_G}(k) = \begin{bmatrix} w\_g_{11}(k) & \dots & w\_g_{1n}(k) \\ \vdots & \ddots & \vdots \\ w\_g_{n1}(k) & \dots & w\_g_{nn}(k) \end{bmatrix},$$

where  $w\_g_{ij}$  gives the value of the weight that agent  $i$  assigned to the distance from  $i$  to  $j$  with considering the influence of the organization. When  $i=j$ , the value of  $w\_g_{ii} = 0$ . Agent  $i$  can only access the  $i$ -th row.

The weights that related to an agent's idea on the importance of the other agents is represented by matrix  $\mathbf{W\_O}$ :

$$\mathbf{W\_O}(k) = \begin{bmatrix} w_{o_{11}}(k) & \dots & w_{o_{1n}}(k) \\ \vdots & \ddots & \vdots \\ w_{o_{n1}}(k) & \dots & w_{o_{nn}}(k) \end{bmatrix},$$

where  $w_{o_{ij}}$  gives the value of the weight that agent  $i$  assigned to the distance from  $i$  to  $j$  with considering the importance of user  $j$ . When  $i=j$ , the value of  $w_{o_{jj}} = 0$ . In other words, it represents the importance of the other agents' requirement on how much information agent  $i$  should release to agent  $j$ . Agent  $i$  can only access the  $i$ -th row.

The weights that related to an agent's idea on the importance of a given task is represented by matrix  $\mathbf{W\_T}$ :

$$\mathbf{W\_T}(k) = \begin{bmatrix} w_{t_{11}}(k) & \dots & w_{t_{1n}}(k) \\ \vdots & \ddots & \vdots \\ w_{t_{n1}}(k) & \dots & w_{t_{nn}}(k) \end{bmatrix},$$

where  $w_{t_{ij}}$  gives the value of the weight that agent  $i$  assigned to the distance from  $i$  to  $j$  with considering the importance of the requirement of the task.

Figure 17 is a flow diagram showing the process of obtaining the weights for all of the agents. Each agent maintains only those weights that relate to itself. In function block 1701, the first agent chosen. A processing loop is entered at function block 1705 where the chosen agent is asked to provide related weights. Function blocks 1715 indexes the next agent, and decision block 1755 determines whether all agents have been processed.

Figure 18 is a flow diagram showing the process of obtaining the weight value related to a given agent  $i$ . In function block 1801, the other agent

is set as agent 1. A processing loop is entered at function block 1805 where the user  $i$  is asked to input the weight values  $w_{s_{ij}}$ ,  $w_{g_{ij}}$ ,  $w_{o_{ij}}$ , and  $w_{t_{ij}}$ . These are weights that are related to agent  $i$  when it is trying to make decisions during the distance selection process. These values also register the personalities of agent  $i$ . Function block 1815 considers the next agent. Decision block 1855 checks whether all the other agents are considered.

Figure 19 is a flow diagram showing the working process of the system. Function block 1901 first identifies the current event with respect to its user. A processing loop is entered at block 1905 which calculates and set the best distances for all the channels. Details of this process is explained with reference to Figure 20. Decision block 1915 checks whether the event has changed. If the event is not changed, no action is needed. Otherwise, the process loops back to function block 1905 to re-calculate the best distances for all the channels.

Figure 20 is a flow diagram showing online distance selection for a given event  $e$ . Function block 2001 chooses the first user for distance calculation. This is because that the agent of the current user needs to determine the distance from the current user to all the other users respectively, and this has to be done one by one. A processing loop is entered at function block 2005 where the best distances for all the channels is calculated. The description of Figure 21 will explain this more. Function block 2015 sets the selected distances for all the channels for the current agent in consideration. Decision block 2055 checks whether there are agents that have not been considered. If so, function block 2075 moves to the next agent for which the distance has not been assigned.

Figure 21 is a flow diagram showing the best distance calculation. Function block 2105 retrieves all the weights related to the current user. Please note that the given agent can only access those weights related to its user. Function block 2115 retrieves the ideal distance and constants. Function

block 2155 calculates the best distance. It uses spring model to find the distance that minimizes the potential energy. The theory goes as follows. Suppose that the task right now is to determine the distance of channel  $k$  from agent  $i$  to agent  $j$ . Suppose that  $x$  is the ideal distance.

$$5 \quad \psi = w_s \times \gamma_s(x, k) + w_g \times \gamma_g(x, k) + w_o \times \gamma_o(x, k) + w_t \times \gamma_t(x, k),$$

where

$$\gamma_s(x, k) = \int_x^{s_{i,j}(k,e)} [k - s_{i,j}(k) \times y] dy = \frac{1}{2} k - s_{i,j}(k) [x - s_{i,j}(k,e)]^2,$$

$$\gamma_g(x, k) = \int_x^{g_{i,j}(k,e)} [k - g_{i,j}(k) \times y] dy = \frac{1}{2} k - g_{i,j}(k) [x - g_{i,j}(k,e)]^2,$$

$$\gamma_o(x, k) = \int_x^{o_{i,j}(k,e)} [k - o_{i,j}(k) \times y] dy = \frac{1}{2} k - o_{i,j}(k) [x - o_{i,j}(k,e)]^2, \text{ and}$$

$$10 \quad \gamma_t(x, k) = \int_x^{t_{i,j}(k,e)} [k - t_{i,j}(k) \times y] dy = \frac{1}{2} k - t_{i,j}(k) [x - t_{i,j}(k,e)]^2.$$

Thus, it is easy to know that

$$15 \quad 2\psi = w_s \times k - s_{i,j}(k) [x^2 - 2xs_{i,j}(k,e) + s_{i,j}(k,e)^2] + \\ w_g \times k - g_{i,j}(k) [x^2 - 2xg_{i,j}(k,e) + g_{i,j}(k,e)^2] + \\ w_o \times k - o_{i,j}(k) [x^2 - 2xo_{i,j}(k,e) + o_{i,j}(k,e)^2] + \\ w_t \times k - t_{i,j}(k) [x^2 - 2xt_{i,j}(k,e) + t_{i,j}(k,e)^2]$$

Obviously, when

$$x = \frac{w_s \times k - s_{i,j}(k,e) + w_g \times k - g_{i,j}(k) \times g_{i,j}(k) + w_o \times k - o_{i,j}(k) + w_t \times k - t_{i,j}(k) \times t_{i,j}(k)}{w_s \times k - s_{i,j}(k) + w_g \times k - g_{i,j}(k) + w_o \times k - o_{i,j}(k) + w_t \times k - t_{i,j}(k)}$$

the potential energy reaches its minimum. Thus,

$$\frac{w-s \times k - s_{i,j}(k,e) + w-g \times k - g_{i,j}(k) \times g_{i,j}(k) + w-o \times k - o_{i,j}(k) + w-t \times k - t_{i,j}(k) \times t_{i,j}(k)}{w-s \times k - s_{i,j}(k) + w-g \times k - g_{i,j}(k) + w-o \times k - o_{i,j}(k) + w-t \times k - t_{i,j}(k)} \text{ is}$$

the value of the distance selected by function block 2155.

In most application applications, it is difficult to provide stiffness functions and to calculate the back-to-ideal potential energies as described above. Furthermore, it might also be difficult to compare different distances given the multi-model nature of the system. In order to avoid these difficulties, we propose a method that uses a set of back-to-ideal energy difference vectors and matrices to guide agents in the selection of distances.

Suppose there are a total of  $M$  different ways to expose one user's status to another user. These  $M$  different ways correspond to  $M$  different distances  $d_1, \dots, d_M$  among users. From a certain point of view, these distances encode the  $Z$  different virtual walls among team members. Suppose that there are a total of  $Q$  different events to be concerned with respect to users in the system.

The back-to-ideal potential energy matrix from  $i$  to  $j$  with respect to user  $i$ ,  $\mathbf{H}_{i,j}^i$ , is given

$$\mathbf{H}_{i,j}^i = \begin{bmatrix} h_{11}^i & \dots & h_{1M}^i \\ \vdots & \ddots & \vdots \\ h_{Q1}^i & \dots & h_{QM}^i \end{bmatrix},$$

where  $h_{uv}^i$  gives the back-to-ideal potential energy when user  $i$  is at event  $u$  and agent  $i$  selected distance  $d_v$  as the distance from  $i$  to  $j$ . If distance  $d_v$  happens to be the ideal distance from  $i$  to  $j$  under event  $u$  with respect to agent

*i*, then  $h_{uv}^i = 0$ . in general, although user *i* might be at different states, only some special events might have different ideal distances. In most situations, user *i*'s ideal distance will be the same, Matrix  $\mathbf{H}_{i,j}^i$  is available to agent *i* at the beginning and is specified by user *i*. The values of the elements of  $\mathbf{H}_{i,j}^i$  encode the degrees of frustrations or tensions user *i* has for different selected distances under different events.

Figure 22 is the flow diagram showing the process of obtaining the values of the Matrix  $\mathbf{H}_{i,j}^i$  for all the *i* and *j*. We set the frustration range to be between 0 and 100. The higher the frustration, the higher the value will be. Suppose that there are totally *E* events to be considered for any agent. The process is initialized by setting *i* to the first agent in function block 2201 and setting *j* to the first agent in function block 2205. A processing loop is entered in function block 2207 where the value for  $\mathbf{H}_{i,j}^i$  is obtained. A determination is made in decision block 2209 as to whether *i* is bigger than the total number of agents, *n*. If not, the index *i* is incremented by 1 in function block 2277, and then the process loops back to function block 2207. If *i* is bigger than *n*, then the index *i* is reset to 1 in function block 2211. A determination is made in decision block 2219 as to whether *j* is bigger than *n*. If so, obtained all the matrices have been obtained; otherwise, the index *j* is incremented by 1 in function block 2215, and the process loops back to function block 2207.

Figure 23 is a flow diagram showing the process of obtaining the value of  $h_{uv}$ . Suppose the *M* different ways of reveal one user's status to another user's status are:  $d_1, \dots, d_M$ . Suppose that there are totally *E* different events that are of interests of the users within the system. The process is initialized by setting *u* to be the first event in function block 2301 and setting *v* to be the

first way of reveal a user's status in function block 2305. A processing loop is entered in function block 2307 where the value of  $h_{uv}$  is input. This is done by asking user  $i$  the question, "what will be the frustration value for user  $i$  if, under the event  $u$ , the information of user  $i$  revealed to user  $j$  will be in the status of  $d_v$ ". User  $i$  will input a frustration value between 0 and 100. a determination is made in decision block 2309 as to whether  $v$  is equal to or bigger than  $M$ . If not,  $v$  is incremented by 1 in function block 2319, and then the process loops bact to function block 2307. If  $v$  is bigger than  $M$ ,  $v$  is reset to 1 in function block 2311. A determination is made in decision block 2315 as to whether  $u$  is equal to or bigger than  $E$ . If not,  $u$  is incremented by 1 in function block 2355, and then the process loops back to function block 2307; otherwise, the process is completed.

The back-to-ideal potential energy vector from  $i$  to  $j$  with respect to the organizational structure is given by  $H_U^{su} = (h_1^{su}, \dots, h_M^{su})$ , where  $h_v^{su}$  gives the back-to-ideal potential energy with respect to the organization when agent  $i$  selects  $d_v$  as the distance from  $i$  to  $j$ . If  $h_v^{su} = 0$ , then  $d_v$  is the ideal distance.

The vector  $H_U^{su}$  is provided by the organization to agent  $i$  at the beginning.

Thus the calculation of  $\delta_2(s_1, \dots, s_M)$  is avoided during the run time.

Figure 24 is a flow diagram showing the process of obtaining the value of  $H_U^{su} = (h_1^{su}, \dots, h_M^{su})$ . The process is initialized by setting  $r$  to 1 in function block 2501. Then a processing loop is entered at function block 2405 where the value of  $h_v^{su}$  is obtained. This is done by inputting the frustration from the organization point of view if the information of user  $i$  revealed to user  $j$  is in the form of  $d_r$ . The value of  $r$  is incremented in function block 2407. Then a determination is made in decision block 2415 as to whether  $r$  is bigger than  $M$ .

If not, the process loops back to function block 2405; otherwise, the value of  $H_U^{su}$  is obtained. Similarly, this process should be executed for every different pair of  $i$  and  $j$ .

The back-to-ideal potential energy vector from  $i$  to  $j$  with respect to agent  $j$  is given by  $H_U^j = (h_1^j, \dots, h_M^j)$ , where  $h_v^j$  gives the back-to-ideal potential energy with respect to agent  $j$  when agent  $i$  selects  $d_v$  as the final distance. This vector encodes agent  $j$ 's preference on distances and is given by user  $j$  to agent  $j$  and is then passed by agent  $j$  to agent  $i$ . The calculation of  $\delta_3(s_1, \dots, s_M)$  is thus avoided.

Figure 25 is a flow diagram showing the process of obtaining the value of  $H_{i,j}^j = (h_1^j, \dots, h_M^j)$ . The process is initialized by setting  $v$  to 1 in function block 2501. A processing loop is entered in function block 2505 where user  $j$  is asked for the value of  $h_v^j$ . The value of  $v$  is incremented by one in function block 2507. Then a determination is made in decision block 2509 as to whether  $v$  is bigger than  $M$ . If not, the process loops back to function block 2505. Otherwise,  $H_{i,j}^j$  is obtained. This process should be executed for every different pair of  $i$  and  $j$ .

The back-to-ideal potential energy vector from  $i$  to  $j$  with respect to a given task  $t_q$  is given by  $H_U^{t_q} = (h_1^{t_q}, \dots, h_M^{t_q})$ , where  $h_v^{t_q}$  gives the back-to-ideal potential energy when agent  $i$  selects  $d_v$  as the distance from  $i$  to  $j$ . The awareness requirements for a collaborative task might be given by the authority who assigns the task, or by the group conventions about the awareness level of the task, or by the system according to various experiences inputted by users. In general, the system divides collaborative tasks into different categories according to the degree of awareness requirements for

each member. It stores these tasks and the associated back-to-ideal potential energy vectors in a common place such that each agent can retrieve the corresponding vector according to its role in the team. The potential energy vectors for all the tasks are available at the beginning, thus the calculation of  $\delta_4(s_1, \dots, s_M)$  is avoided.

Figure 26 is a flow diagram showing the process of obtaining the value of  $H_U^{t_q} = (h_1^{t_q}, \dots, h_M^{t_q})$ . The process is initialized by setting  $v$  to 1 in function block 2601. A processing loop is entered in function block 2605 where the value of  $h_v^{t_q}$  is retrieved for the given task. The value of  $v$  is incremented by 1 in function block 2607. Then a determination is made in decision block 2609 as to whether  $v$  is bigger than  $M$ . If not, the process loops back to function block 2605; otherwise,  $H_i^{t_q}$  is obtained. This process should be executed for every different pair of  $i$  and  $j$ , and for every task  $t_q$ .

As discussed above, the related back-to-ideal potential energies are all available for agent  $i$ . Thus, when a new collaboration task is assigned to user  $i$  or a new event is happening to user  $i$ , agent  $i$  will update the distances from its user to all the other related users.

Suppose that at time  $\tau$ , user  $i$  is at the state of event  $u$  and the current collaboration task is  $t_q$ , then the weighted back-to-ideal potential energies for distance  $d_v$  is  $\delta_v = w_U^i \times h_{uv}^i + w_U^{su} \times h_v^{wu} + w_U^j \times h_v^j + w_U^{ask} \times h_v^{t_q}$ . To select the best distance, agent  $i$  calculates the weighted back-to-ideal potential energies  $\delta(d_1), \dots, \delta(d_M)$  for all the distances  $d_1, \dots, d_M$  and chooses the distance  $d$  with the minimum energy as the value of  $d_U(\tau)$ , the distance from  $i$  to  $j$  at time  $\tau$ . In other words, if  $\delta(d) \leq \delta(d_v)$  ( $v=1, \dots, M$ ), then  $d_U(\tau) = d$ .

At the beginning, all the distances within the system select their awareness distances to all the other agents according to the above method by

assuming that there is no collaboration task. Thus, only the first three terms are involved in the calculation:  $\delta_v = w_U^i \times h_{uv}^i + w_U^{su} \times h_v^{su} + w_U^j \times h_v^j$ . After  $\Psi(0)$  is determined, if there is no change in the status of any users and there is no new task, then the awareness status of the system will stay the same. This status will be updated whenever there are changes in events or tasks. When a change occurs, each related agent will update its distances to all the other agents according to the above described method. The awareness status  $\Psi(\tau)$  of the system is a system that is adaptive to events and tasks. Each element  $d_U(\tau)$  of  $\Psi(\tau)$  is an adaptive media wall in the virtual organization of the system. It is these virtual walls that keep the organization functioning and provide adaptive awareness to all the members of the team.

In the event that the spring model solution is difficult to obtain, a mixed approach of the spring model and the following mechanism may be used. A matrix and vector look up model can be used to determine the distances among distributed users. The values of the matrix and the vector encodes the preferences of the user, the preferences of the task, the preferences of the organization, and the preference requirements of the other user who receives the awareness information.

While the invention has been described in terms of a single preferred embodiment, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.